

Personalisasi Modul Similaritas Pencarian Lowongan Kerja dengan Algoritma *Extended Weighted Tree Similarity*

¹⁾Tjiong Debora Yulianti, ²⁾Ade Iriani, ³⁾Hendry

Fakultas Teknologi Informasi

Universitas Kristen Satya Wacana

Jl. Diponegoro 52-60, Salatiga 50771, Indonesia

email: ¹⁾ih_aneh@yahoo.co.id, ²⁾adeiriani@gmail.com ³⁾hendry.honk@gmail.com

Abstract

Extended Weighted Tree Similarity is an algorithm used to get the similarity level between two objects by multiplying and calculating the weight of the objects. The result is determined by the range between zero to one, and the result that closer to one has the highest similarity level. In this study, extended weighted tree similarity used to apply search system of jobs by using Indonesian language as the input. Extended weighted tree similarity used to compare user input to web page and find the value of similarity. As a result, this system will display the web page from the highest rank to the lowest rank for the similarity method. This system compares the result of search by AND method, OR method, and extended weighted tree similarity then displayed in a graph. Based on the result of graph, the similarity method is between AND method and OR method. Similarity is better than other method, because the result from search process not too wide and narrow and the information that is served not too strict and not too loose.

Keywords : Extended Weighted Tree Similarity, Jobs, AND, OR.

1. Pendahuluan

Sumber daya manusia merupakan suatu faktor penting dalam menunjang kemajuan suatu organisasi. Hal ini disadari seiring dengan adanya perubahan lingkungan bisnis serta perubahan teknologi yang cepat, yang menyebabkan terjadinya peningkatan permintaan tenaga kerja dengan *skill* yang lebih baik. Perubahan yang terjadi membawa dampak terhadap peranan manajemen SDM yang awalnya hanya fokus pada fungsi produksi, keuangan, serta pemasaran menjadi lebih terlibat aktif dalam perencanaan, pengelolaan, serta pengendalian organisasi yang berkaitan dengan alokasi dan pengembangan SDM. Manajemen SDM dituntut untuk mampu bersaing di tengah perkembangan yang sedang dan akan terjadi dengan menempatkan SDM yang produktif, berkualitas, serta cepat beradaptasi dengan perubahan-perubahan yang ada [1]. Kondisi tersebut menyebabkan lowongan kerja sulit untuk diperoleh, karena adanya proses seleksi yang dilakukan untuk menentukan tenaga kerja yang sesuai dengan lowongan kerja yang tersedia. Faktor lain yang menyebabkan lowongan kerja sulit diperoleh yaitu lowongan kerja yang tersedia masih terbatas jika

dibandingkan jumlah tenaga kerja yang ada. Seiring dengan perkembangan teknologi yang ada, proses pencarian lowongan kerja dapat dilakukan dengan banyak cara, salah satunya yaitu melalui internet. Pencarian informasi lowongan kerja melalui internet sangat ditentukan oleh *keywords* yang diinputkan. Metode yang digunakan dalam proses pencarian melalui internet dibagi menjadi dua, yaitu metode pencarian *logical AND* dan metode pencarian *logical OR*. Pencarian *logical AND* akan menghasilkan informasi yang sifatnya terbatas, karena pencarian ini akan mencari semua *keywords* yang ada. Pencarian tidak berhasil dan informasi tidak dapat ditampilkan jika satu *keyword* tidak dapat ditemukan. Pencarian *logical OR* akan menghasilkan informasi yang sifatnya terlalu melebar, karena pencarian dilakukan dengan mencari setiap *keyword* yang ada [2]. Metode pencarian *logical AND* dan *OR* membuat informasi yang diberikan menjadi kurang tepat atau tidak relevan dengan input-an pencarian *full-text*, yang dalam hal ini mengenai lowongan kerja. Berdasarkan permasalahan tersebut, maka penelitian ini akan dirancang suatu modul pencarian informasi lowongan kerja dengan menggunakan algoritma *Extended Weighted Tree Similarity*, di mana proses pencarian dilakukan bukan dengan ditemukannya *input-an user*, melainkan dengan melakukan perhitungan tingkat kemiripan halaman *web* dengan *input-an user*. Penelitian ini berkaitan dengan *Information Retrieval*, di mana pada proses pencarian dengan metode similaritas, terdapat proses *string processing* untuk mengambil isi dari suatu halaman *web* dan dilakukan proses *stemming* kata bahasa Indonesia menggunakan algoritma Nazief dan Adriani. Penelitian yang dilakukan bersifat *offline* dan akan dilakukan perbandingan hasil pencarian lowongan kerja dengan menggunakan metode *AND*, metode *OR*, serta metode similaritas.

2. Kajian Pustaka

Struktur morfologi bahasa Indonesia terdiri dari *Inflectional Word* merupakan akhiran yang tidak mempengaruhi arti dasar dari sebuah kata asal. *Inflectional suffix* terdiri dari dua, yaitu : *Particle* (P), yang ditujukan untuk melakukan penegasan atau penekanan, contohnya “-lah”, “-kah”, “-pun”, “-tah” ; *Possessive Pronouns* (PP), yaitu kata ganti yang menunjukkan kepunyaan, contoh “-ku”, “-mu”, “-nya”. *Derivational Word* meliputi *Derivational Prefix*, yaitu “ber-”, “di-”, “ke-”, “meng-”, “peng-”, “per-”, “ter-” ; *Derivational Suffix*, yaitu “-i”, “-kan”, “-an” ; dan *Derivational Confix*, merupakan kombinasi antara *prefix* dan *suffix* yang ditambahkan secara bersama-sama untuk membentuk kata turunan baru. Kombinasi *prefix* dan *suffix* tidak semuanya diperbolehkan. Tabel 1[3] merupakan tabel kombinasi *prefix* dan *suffix* yang tidak diijinkan.

Tabel 1. Tabel Kombinasi *Prefix* dan *Suffix* yang Tidak Diiijinkan[3]

<i>Prefix</i>	<i>Suffix</i>
ber-	-i
di-	-an
ke-	-i, -kan
meng-	-an
peng-	-i, -kan
ter-	-an

Algoritma Nazief dan Adriani merupakan algoritma *stemming* kata berbahasa Indonesia. Algoritma ini menggunakan alat bantu berupa kamus kata dasar, yang akan dicek pada setiap akhir langkah *stemming* [4].

1. Mencari kata yang akan distem dalam kamus, jika ditemukan maka kata tersebut diasumsikan sebagai kata dasar, dan algoritma berhenti.
2. Membuang *inflection suffixes* (“-lah”, “-kah”, “-ku”, “-mu”, “-nya”), jika berupa *particles* (“-lah”, “-kah”, “-tah”, atau “-pun”) maka langkah ini diulangi lagi untuk menghapus *possessive pronouns* (“-ku”, “-mu”, atau “-nya”) yang ada.
3. Menghapus *derivational suffixes* (“-i”, “-an”, atau “-kan”), jika berhasil maka lakukan langkah empat. Langkah ini dilakukan jika langkah empat gagal :
 - a. Dilakukan pengecekan, jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k” maka “-k” juga ikut dihapus, dan lakukan kembali langkah empat. Lakukan langkah b jika gagal.
 - b. Akhiran yang dihapus (“-i”, “-an”, atau “-kan”) dikembalikan.
4. Hapus *derivation prefix*. Lakukan langkah a jika ada sufiks yang dihapus pada langkah tiga. Lakukan langkah b jika tidak ada sufiks yang dihapus.
 - a. Periksa tabel kombinasi awalan-akhiran yang tidak diijinkan. Algoritma berhenti jika kombinasi ditemukan. Lakukan langkah b jika kombinasi tidak ditemukan.
 - b. Dilakukan perulangan hingga tiga kali untuk menentukan tipe awalan, kemudian menghapus awalan. Dilakukan pengecekan jika awalan kedua sama dengan awalan pertama atau tiga prefiks telah dihapus, maka algoritma berhenti.
5. Melakukan pengecekan, apabila kata dasar tidak ditemukan, maka dilakukan *recoding* yaitu mengembalikan huruf awal yang telah dihapus dari akar kata sebelumnya.

- Semua langkah telah selesai dilakukan tetapi tidak juga berhasil menemukan kata dasar, maka kata awal diasumsikan sebagai kata dasar (*root word*). Proses selesai.

Langkah-langkah untuk menentukan tipe awalan [4] :

- Tipe awalan “di-”, “ke-”, atau “se-” secara berturut-turut yaitu “di-”, “ke-”, atau “se-”.
- Dibutuhkan sebuah proses tambahan untuk menentukan tipe awalan jika awalannya “te-”, “me-”, “be-”, atau “pe-”.
- Langkah berhenti jika dua karakter pertama bukan “di-”, “ke-”, “se-”, “te-”, “be-”, “me-”, atau “pe-”.
- Langkah berhenti jika tipe awalan adalah “none”, namun jika tipe awalan adalah bukan “none” maka awalan dapat dilihat pada Tabel 2[4]. Hapus awalan jika ditemukan.

Tabel 2. Cara Menentukan Tipe Awalan untuk Kata yang Diawali dengan “te-“ [4]

Following characters				Tipe
Set 1	Set 2	Set 3	Set 4	awalan
“-r-”	“-r-”	-	-	none
“-r-”	vowel	-	-	ter-luluh
“-r-”	not (vowel or “-r-”)	“-er-”	vowel	ter
“-r-”	not (vowel or “-r-”)	“-er-”	not vowel	ter-
“-r-”	not (vowel or “-r-”)	not “-er-”	-	ter
not (vowel or “-r-”)	“-er-”	vowel	-	none
not (vowel or “-r-”)	“-er-”	not vowel	-	te

Tabel 3 [4] menunjukkan jenis awalan yang harus dihapus berdasarkan tipe awalannya.

Tabel 3. Jenis Awalan Berdasarkan Tipe Awalannya [4]

Tipe awalan	Awalan yang harus dihapus
di-	di-
ke-	ke-
se-	se-
te-	te-

ter-	ter-
ter-luluh	ter

Berikut merupakan aturan-aturan tambahan dalam algoritma ini [4] :

1. Aturan untuk reduplikasi dilakukan dengan pengecekan, jika kedua kata yang dihubungkan oleh kata penghubung adalah kata yang sama, maka yang menjadi kata dasar adalah bentuk tunggalnya, contoh pada kata “buku-buku”, maka kata dasarnya yaitu “buku”. Sedangkan kata lain seperti “bolak-balik”, “berbalas-balasan”, dan “seolah-olah” untuk mendapatkan kata dasarnya, kedua kata diartikan secara terpisah. Jika keduanya memiliki kata dasar yang sama maka diubah menjadi bentuk tunggal, contoh kata “berbalas-balasan”, “berbalas”, dan “balasan” memiliki kata dasar yang sama yaitu “balas”, sehingga kata dasar “berbalas-balasan” adalah “balas”. Sebaliknya, kata “bolak-balik”, “bolak” dan “balik” memiliki kata dasar yang berbeda, maka kata dasarnya yaitu “bolak-balik”.
2. Tambahan bentuk awalan dan akhiran serta aturannya. Tipe awalan “mem-”, kata yang diawali dengan awalan “memp-” memiliki tipe awalan “mem-”. Tipe awalan “meng-”, kata yang diawali dengan awalan “mengk-” memiliki tipe awalan “meng”.

Extended Weighted Tree Similarity, merupakan algoritma yang digunakan untuk menentukan tingkat kemiripan antar dua objek, dengan mengalikan serta mengkalkulasikan nilai bobot yang dimiliki objek tersebut. Nilai hasil perhitungan bernilai antara *range* nol hingga satu, semakin nilai mendekati satu maka objek tersebut memiliki tingkat kemiripan yang tinggi. Algoritma ini merepresentasikan objek ke dalam bentuk *tree* dan XML [5]. Suatu *tree* terdiri dari *parent* yang dapat memiliki atribut berupa bobot, dan nama *parent*. Masing-masing *parent* dapat memiliki *identifier*. Suatu *identifier* juga memiliki bobot sebagai atribut serta memiliki isi. Total bobot *parent* bernilai satu, untuk memperoleh bobot suatu *parent*, digunakan Persamaan 1 [6].

$$W_i = 1/n \quad (1)$$

Keterangan :

W_i : bobot *parent* ke- i
 n : total *parent* yang ada

Bobot suatu *identifier* diperoleh dari Persamaan 2 [6].

$$W_{ind} = \text{frek}/n \quad (2)$$

Keterangan :

W_{ind} : bobot *identifier*
 $frek$: jumlah kemunculan *identifier*
 n : total *identifier* yang ada

Gambar 1 merupakan *tag* umum dalam membentuk representasi XML.

```
<cterm>
    <_r ...>
        <ind ..> ... </ind>
        <ind ..> ... </ind>
    </_r>
</cterm>
```

Gambar 1. Tag Representasi XML

<cterm> dan </cterm> merupakan *tag* untuk mengawali serta mengakhiri suatu *tree*. <_r> merupakan *tag parent* dari suatu *tree*, yang dapat memiliki atribut. Setiap *parent* dapat memiliki beberapa *identifier*, yang direpresentasikan dalam *tag* <ind> yang juga dapat memiliki atribut. Persamaan 3[6] merupakan rumus perhitungan tingkat kemiripan antara dua objek yang dibandingkan.

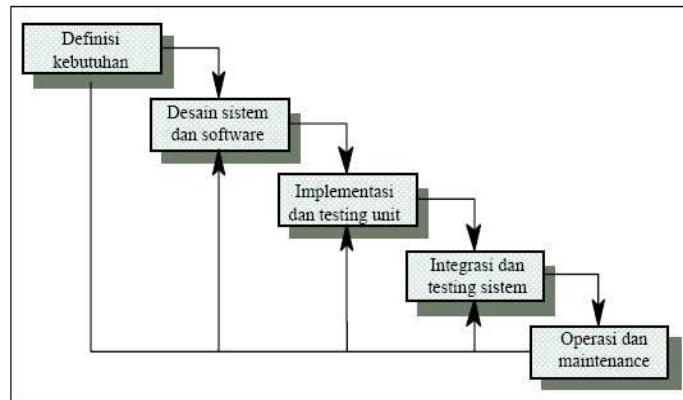
$$BK = \sum((W_i * W_j) * (\sum(W_{ind(i)} * W_{ind(j)}))) \quad (3)$$

Keterangan :

BK : bobot kemiripan
 W_i : bobot *parent* halaman *web* ke-*i*
 W_j : bobot *parent input-an user* ke-*j*
 $W_{ind(i)}$: bobot *identifier* halaman *web* ke-*i*
 $W_{ind(j)}$: bobot *identifier input-an user* ke-*j*

3. Metode Penelitian

Metode yang digunakan dalam penerapan sistem yaitu metode *waterfall*. Gambar 2[7] merupakan alur metode *waterfall*.

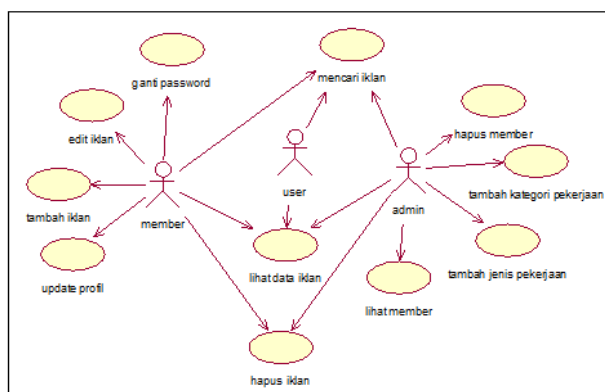


Gambar 2. Metode Waterfall [7]

1. Tahap yang pertama yaitu definisi kebutuhan, di mana pada tahap ini dilakukan analisis kebutuhan yang harus dipenuhi oleh sistem, yang dalam hal ini yaitu sistem pencarian lowongan kerja.
2. Tahap kedua yaitu perancangan sistem. Tahapan ini akan menghasilkan suatu gambaran aplikasi pencarian lowongan kerja sesuai dengan kebutuhan yang harus dipenuhi.
3. Tahap ketiga merupakan tahap implementasi serta pengujian, di mana dilakukan pada tahap ini dilakukan pembuatan aplikasi pencarian lowongan kerja, serta dilakukan pengujian dari setiap bagian dari sistem.
4. Tahapan keempat merupakan tahap pengujian sistem secara keseluruhan serta dilakukan evaluasi apakah aplikasi pencarian lowongan kerja yang dibangun memiliki manfaat serta memudahkan pengguna dalam mencari lowongan kerja.
5. Tahap yang terakhir, yaitu tahap pengoperasian dan pemeliharaan sistem. Tahapan ini merupakan tahap perawatan sistem yang telah dibangun serta dilakukan perbaikan apabila ditemukan kesalahan.

3.1. Use Case Diagram

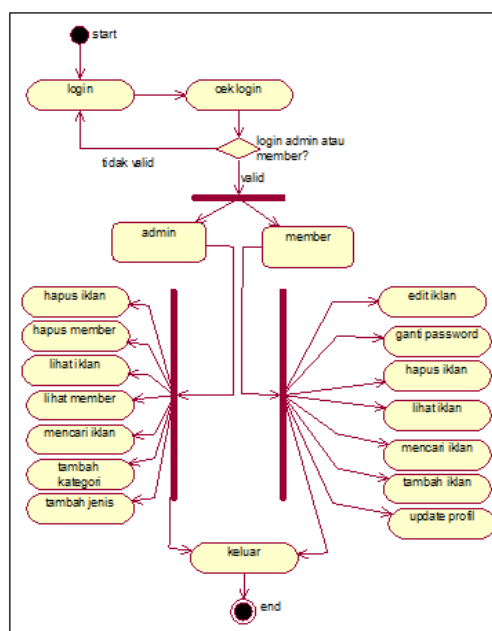
Gambar 3 merupakan penggambaran *use case diagram* sistem, di mana aktor dalam sistem terdiri dari tiga, yaitu *member*, *user*, dan *admin*. Aktor *member* adalah aktor yang melakukan registrasi dalam sistem, yang dapat melakukan pemasangan iklan, yang dalam hal ini adalah berupa perusahaan atau instansi. *User* adalah pengguna yang tidak menjadi anggota sistem, sehingga aksesnya terbatas, yaitu mencari iklan dan melihat data iklan. *Member* mempunyai tujuh hak akses, yaitu hapus iklan, lihat iklan, *edit* iklan, tambah iklan, ganti *password*, dan *update* profil, dan mencari iklan. Hak akses *admin* terdiri dari hapus *member*, lihat *member*, tambah jenis pekerjaan, tambah kategori pekerjaan, hapus iklan, melihat data iklan, dan mencari iklan.



Gambar 3. Use Case Diagram Sistem

3.2. Activity Diagram

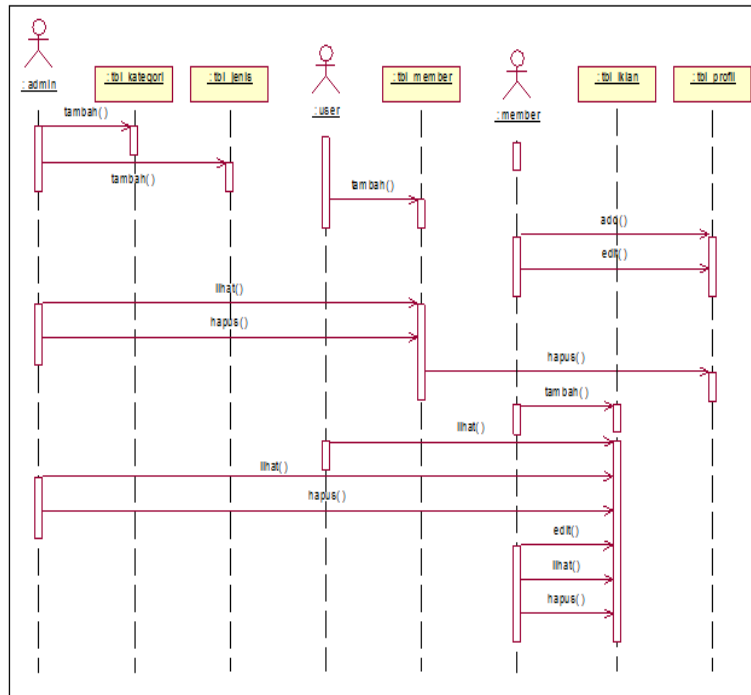
Gambar 4 merupakan gambar *activity diagram* sistem, di mana langkah yg dilakukan untuk dapat mengakses sistem yaitu melakukan *login*. *Login* akan dicek, apakah *login* sebagai *admin* atau *member*, jika *login* salah maka kembali *login*, tapi jika *valid*, maka akan tampil menu-menu aktivitas sesuai dengan *login* yang dilakukan.



Gambar 4. Activity Diagram Sistem

3.3. Sequence Diagram

Gambar 5 merupakan menunjukkan *sequence diagram* sistem yang terdiri dari tiga aktor, yaitu *user*, *member* dan *admin*. *User* bisa melakukan registrasi agar dapat menjadi *member* dan mengakses sistem, seperti menambah iklan atau hak akses lainnya.

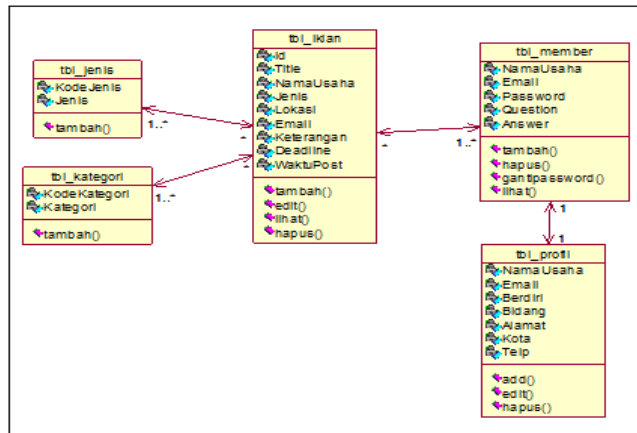


Gambar 5. Sequence Diagram Sistem

3.4. Class Diagram

Gambar 6 merupakan menunjukkan *class diagram* sistem, di mana menunjukkan hubungan antara *tbl_jenis*, *tbl_kategori*, *tbl_iklan*, *tbl_member*, dan *tbl_profil* dan penggambaran atribut-atribut dan operasi yang dimiliki. *Tbl_jenis* mempunyai dua atribut, yaitu *KodeJenis* dan *Jenis* dan operasi *tambah()*. *Tbl_kategori* mempunyai dua atribut, yaitu *KodeKategori* dan *Kategori* dan mempunyai operasi *tambah()*. *Tbl_iklan* mempunyai atribut *id*, *Title*, *NamaUsaha*, *Jenis*, *Lokasi*, *Email*, *Keterangan*, *Deadline*, *WaktuPost*, dan mempunyai operasi *tambah()*, *edit()*, *lihat()*, dan *hapus()*. *Tbl_member* mempunyai atribut *NamaUsaha*, *Email*, *Password*, *Question*, *Answer* dan mempunyai operasi *tambah()*, *hapus()*, *gantipassword()*, *lihat()*. *Tbl_profil*

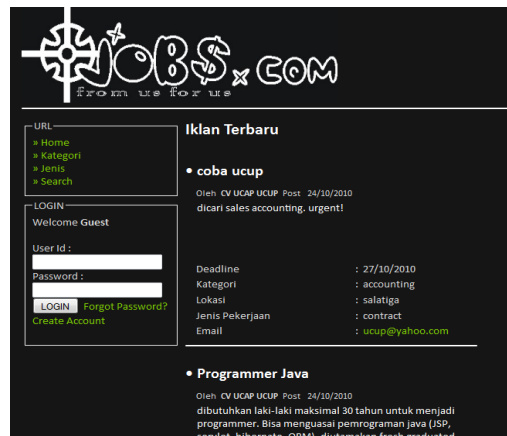
mempunyai atribut NamaUsaha, Email, Berdiri, Bidang, Alamat, Kota, Telp dan operasi *add()*, *edit()*, *hapus()*.



Gambar 6. Class Diagram Sistem

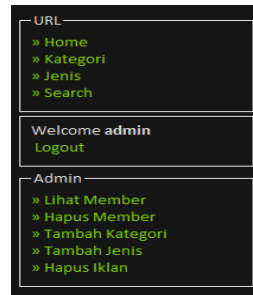
4. Implementasi Sistem

Sistem dapat diakses oleh dua pengguna, yaitu *admin* dan *member*. *Member* merupakan suatu perusahaan atau instansi yang dapat melakukan registrasi sehingga dapat melakukan penambahan iklan. Gambar 7 merupakan tampilan awal sistem. Tampilan awal sistem akan menampilkan iklan-iklan terbaru yang ditambahkan oleh *member*. Pada bagian sebelah kiri terdapat menu-menu seperti *link Home*, *link Kategori*, *link Jenis*, *link Search* serta terdapat *login*. Pada *login* terdapat *link Create Account* yang berfungsi untuk registrasi *member* dan *link Forgot Password* yang berfungsi apabila *member* lupa *password login*.



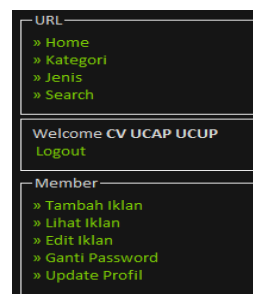
Gambar 7. Tampilan Awal Sistem

Link Home berfungsi untuk mengarahkan pada tampilan awal sistem. *Link Kategori* merupakan *link* yang menampilkan kategori-kategori yang ada dalam *database* yang dapat dipilih untuk menampilkan iklan sesuai kategori tertentu. *Link Jenis* berfungsi menampilkan jenis pekerjaan yang dapat dipilih untuk menampilkan iklan berdasarkan jenis pekerjaan tersebut. *Link Search* berfungsi untuk melakukan pencarian lowongan kerja. Gambar 8 merupakan menu-menu yang dapat diakses oleh *admin*.



Gambar 8. Menu *Admin*

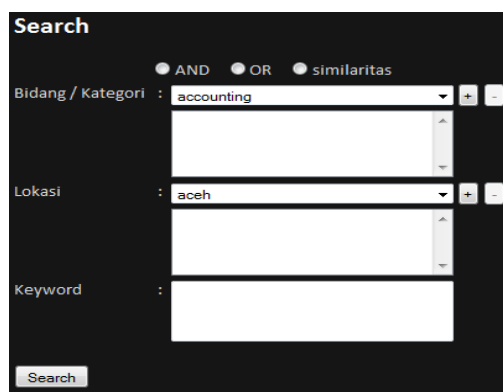
Link Lihat Member berfungsi untuk melihat data semua *member* yang terdaftar. *Link Hapus Member* berfungsi untuk menghapus data *member* tertentu, penghapusan data dilakukan dengan memilih *email member* yang ingin dihapus. *Link Tambah Kategori* merupakan *link* untuk menambah kategori pekerjaan dengan menginputkan nama kategori. *Link Tambah Jenis* berfungsi untuk menambah jenis pekerjaan dengan memasukkan jenis pekerjaan. *Link Hapus Iklan* merupakan *link* yang berfungsi untuk menghapus iklan berdasarkan *deadline* yang telah berakhir. Gambar 9 merupakan menu yang dapat diakses oleh *member*.



Gambar 9. Menu *Member*

Link Tambah Iklan berfungsi untuk menambahkan data iklan lowongan kerja. *Link Lihat Iklan* berfungsi untuk melihat data iklan lowongan kerja yang

dimiliki *member* tersebut, sekaligus terdapat *link delete* untuk menghapus data iklan tertentu. *Link Edit* Iklan berfungsi untuk melakukan perubahan terhadap data iklan lowongan kerja. Perubahan data iklan berdasarkan pada *id* data iklan. *Link Ganti Password* merupakan *link* untuk mengganti *password login*. *Link Update* Profil merupakan *link* untuk mengubah data profil *member*. Tampilan halaman pencarian lowongan ditunjukkan pada Gambar 10.

The image shows a web interface titled "Search". At the top, there are three radio buttons for search methods: "AND" (selected), "OR", and "similaritas". Below this, there are three input fields. The first is labeled "Bidang / Kategori" and contains the text "accounting". The second is labeled "Lokasi" and contains the text "aceh". The third is labeled "Keyword" and is empty. To the right of each input field are small "+" and "-" buttons. At the bottom left of the form is a "Search" button.

Gambar 10. Tampilan Halaman Pencarian

Proses pencarian dilakukan dengan memasukkan *input-an* berupa kategori, lokasi, serta *keyword*. *Input-an* yang harus dimasukkan minimal satu, dan *user* harus memilih metode pencarian yang ingin digunakan. Metode pencarian yang dapat digunakan yaitu metode *AND*, *OR*, serta metode *similaritas*.

5. Hasil Analisis Sistem

Pengujian dilakukan dengan memasukkan total *input-an* mulai dari tiga hingga dua belas *input-an* dengan masing-masing menggunakan metode *AND*, *OR*, dan metode *similaritas*. Tabel 4 merupakan tabel hasil pengujian dengan tiga *input-an*, yaitu kategori *accounting* dan *administration*, lokasi *aceh* dan *keyword* akuntansi. Hasil yang diperoleh yaitu 17 halaman *web* untuk metode *AND*, 113 halaman *web* untuk metode *OR*, 112 halaman untuk metode *similaritas*.

Tabel 4. Hasil Pengujian dengan Tiga *Input-an*

<i>Input-an</i>		Hasil		
		<i>AND</i>	<i>OR</i>	similaritas
Kategori	<i>accounting</i>			
Lokasi	aceh	17	113	112
<i>Keyword</i>	akuntansi			

Tabel 5 merupakan tabel hasil pengujian dengan empat *input-an*, yaitu kategori *accounting* dan *administration*, lokasi aceh dan *keyword* akuntansi. Hasil yang diperoleh yaitu 17 halaman *web* untuk metode *AND*, 113 halaman *web* untuk metode *OR*, 112 halaman untuk metode similaritas.

Tabel 5. Hasil Pengujian dengan Empat *Input-an*

<i>Input-an</i>		Hasil		
		<i>AND</i>	<i>OR</i>	similaritas
Kategori	<i>accounting</i> <i>administration</i>	17	113	112
Lokasi	aceh			
<i>Keyword</i>	akuntansi			

Tabel 6 merupakan tabel hasil pengujian dengan lima *input-an*, yaitu kategori *accounting*, *administration*, lokasi aceh, bali dan *keyword* akuntansi. Hasil yang diperoleh yaitu 17 halaman *web* untuk metode *AND*, 113 halaman *web* untuk metode *OR*, 112 halaman untuk metode similaritas.

Tabel 6. Hasil Pengujian dengan Lima *Input-an*

<i>Input-an</i>		Hasil		
		<i>AND</i>	<i>OR</i>	similaritas
Kategori	<i>accounting</i> <i>administration</i>	17	113	112
Lokasi	aceh bali			
<i>Keyword</i>	akuntansi			

Tabel 7 merupakan tabel hasil pengujian dengan enam *input-an*, yaitu kategori *accounting*, *administration*, lokasi aceh, bali dan *keyword* akuntansi, staff. Hasil yang diperoleh yaitu sembilan halaman *web* untuk metode *AND*, 113 halaman *web* untuk metode *OR*, empat halaman untuk metode similaritas.

Tabel 7. Hasil Pengujian dengan Enam *Input-an*

<i>Input-an</i>		Hasil		
		<i>AND</i>	<i>OR</i>	similaritas
Kategori	<i>accounting</i> <i>administration</i>			
Lokasi	aceh bali	9	113	4
<i>Keyword</i>	akuntansi staff			

Tabel 8 merupakan tabel hasil pengujian dengan tujuh *input-an*, yaitu kategori *accounting*, *administration*, yang ditambahkan *advertising*, lokasi aceh, bali dan *keyword* akuntansi, staff. Hasil yang diperoleh yaitu sembilan halaman *web* untuk metode *AND*, 113 halaman *web* untuk metode *OR*, empat halaman untuk metode similaritas.

Tabel 8. Hasil Pengujian dengan Tujuh *Input-an*

<i>Input-an</i>		Hasil		
		<i>AND</i>	<i>OR</i>	similaritas
Kategori	<i>accounting</i> <i>administration</i> <i>advertising</i>			
Lokasi	aceh bali	9	113	4
<i>Keyword</i>	akuntansi staff			

Tabel 9 merupakan tabel hasil pengujian dengan delapan *input-an*, yaitu kategori *accounting*, *administration*, yang ditambahkan *advertising*, lokasi aceh, bali, yang ditambah jawa barat dan *keyword* akuntansi, staff. Hasil yang diperoleh yaitu lima halaman *web* untuk metode *AND*, 113 halaman *web* untuk metode *OR*, empat halaman untuk metode similaritas.

Tabel 9. Hasil Pengujian dengan Delapan *Input-an*

<i>Input-an</i>		Hasil		
		<i>AND</i>	<i>OR</i>	similaritas
Kategori	<i>accounting</i>			
	<i>administration</i>			
	<i>advertising</i>			
Lokasi	aceh	5	113	4
	bali			
	jawa barat			
<i>Keyword</i>	akuntansi			
	staff			

Tabel 10 merupakan tabel hasil pengujian dengan sembilan *input-an*, yaitu kategori *accounting*, *administration*, *advertising*, lokasi aceh, bali, jawa barat dan *keyword* akuntansi, staff, iklan. Hasil yang diperoleh yaitu lima halaman *web* untuk metode *AND*, 113 halaman *web* untuk metode *OR*, 10 halaman untuk metode similaritas.

Tabel 10. Hasil Pengujian dengan Sembilan *Input-an*

<i>Input-an</i>		Hasil		
		<i>AND</i>	<i>OR</i>	similaritas
Kategori	<i>accounting</i>			
	<i>administration</i>			
	<i>advertising</i>			
Lokasi	aceh	5	113	10
	bali			
	jawa barat			
<i>Keyword</i>	akuntansi			
	staff			
	iklan			

Tabel 11 merupakan tabel hasil pengujian dengan sepuluh *input-an*, yaitu kategori *accounting*, *administration*, *advertising*, *education*, lokasi aceh, bali, jawa barat dan *keyword* akuntansi, staff, iklan. Hasil yang diperoleh yaitu lima halaman *web* untuk metode *AND*, 113 halaman *web* untuk metode *OR*, 10 halaman untuk metode similaritas.

Tabel 11. Hasil Pengujian dengan Sepuluh *Input-an*

<i>Input-an</i>	Hasil		
	<i>AND</i>	<i>OR</i>	similaritas
Kategori			
	<i>accounting</i>		
	<i>administration</i>		
	<i>advertising</i>		
	<i>education</i>		
Lokasi			
	aceh	5	113
	bali		10
	jawa barat		
	akuntansi		
Keyword			
	staff		
	iklan		

Tabel 12 merupakan tabel hasil pengujian dengan sebelas *input-an*, yaitu kategori *accounting*, *administration*, *advertising*, *education*, lokasi aceh, bali, jawa barat, jawa tengah dan *keyword* akuntansi, staff, iklan. Hasil yang diperoleh yaitu lima halaman *web* untuk metode *AND*, 113 halaman *web* untuk metode *OR*, 10 halaman untuk metode similaritas.

Tabel 12. Hasil Pengujian dengan Sebelas *Input-an*

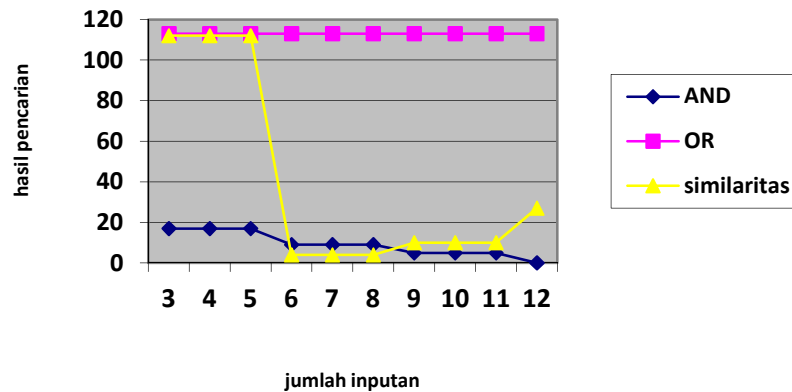
<i>Input-an</i>	Hasil		
	<i>AND</i>	<i>OR</i>	similaritas
Kategori			
	<i>accounting</i>		
	<i>administration</i>		
	<i>advertising</i>		
	<i>education</i>		
Lokasi			
	aceh	5	113
	bali		10
	jawa barat		
	jawa tengah		
	akuntansi		
Keyword			
	staff		
	iklan		

Tabel 13 merupakan tabel hasil pengujian dengan dua belas *input-an*, yaitu kategori *accounting*, *administration*, *advertising*, *education*, lokasi aceh, bali, jawa barat, jawa tengah dan *keyword* akuntansi, staff, iklan, guru. Hasil yang diperoleh yaitu 0 halaman *web* untuk metode *AND*, 113 halaman *web* untuk metode *OR*, 27 halaman untuk metode similaritas.

Tabel 13. Hasil Pengujian dengan Dua Belas *Input-an*

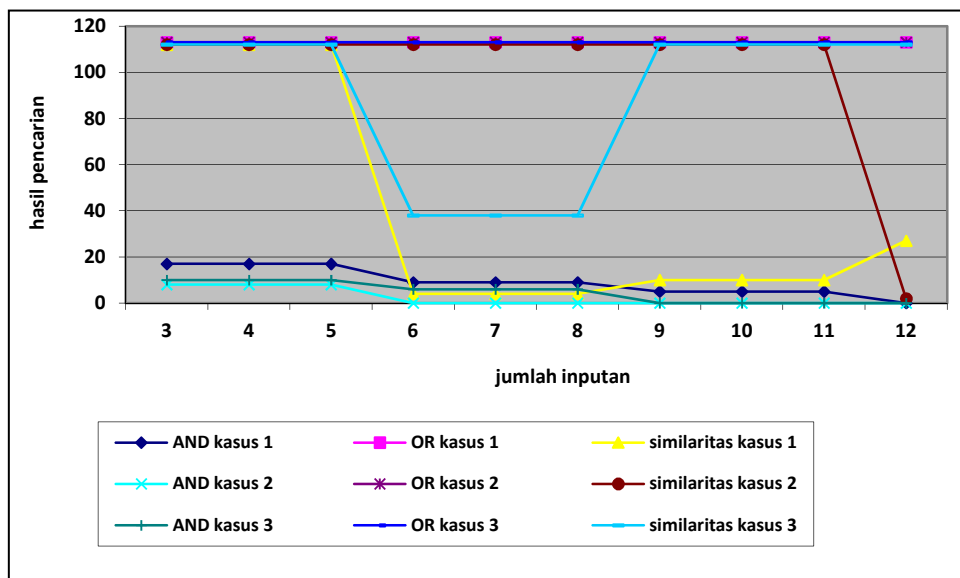
<i>Input-an</i>		Hasil		
		<i>AND</i>	<i>OR</i>	similaritas
Kategori	<i>accounting</i>			
	<i>administration</i>			
	<i>advertising</i>			
	<i>education</i>			
Lokasi	aceh			
	bali			
	jawa barat	0	113	27
	jawa tengah			
<i>Keyword</i>	akuntansi			
	staff			
	iklan			
	guru			

Gambar 11 merupakan grafik hasil pengujian yang telah dilakukan. Berdasarkan grafik, metode *OR* selalu menghasilkan nilai terbanyak, karena jika satu *input-an* saja ditemukan maka halaman tersebut akan ditampilkan. Metode *AND* menghasilkan nilai nol pada *input-an* ke dua belas, karena jika satu *input-an* tidak ditemukan, maka tidak akan ada halaman *web* yang ditampilkan. Metode similaritas menampilkan hanya hasil yang lebih dari atau sama dengan nilai *threshold*, yang diperoleh dari $(\text{nilai tertinggi} + \text{nilai terendah})/2$. Hasil pencarian dengan metode similaritas berada di antara hasil pencarian dengan metode *AND* dan metode *OR*.



Gambar 11. Grafik Hasil Pengujian

Pengujian sistem dilakukan dengan tiga kasus untuk tiga metode, yaitu metode *AND*, *OR*, similaritas, di mana masing-masing kasus diuji dengan tiga hingga dua belas total *input*-an. Gambar 12 merupakan grafik hasil pengujian untuk tiga kasus tersebut. Berdasarkan Gambar 12, hasil pencarian dengan metode similaritas berada di antara pencarian dengan metode *AND* dan metode *OR*. Hasil pencarian dengan metode *AND* selalu berada di paling bawah dan pencarian dengan metode *OR* selalu berada di atas.



Gambar 12. Grafik Hasil Semua Pengujian

6. Simpulan

Berdasarkan hasil pembuatan sistem, dapat disimpulkan bahwa banyaknya *input*-an serta metode yang digunakan mempunyai pengaruh yang besar terhadap hasil pencarian. Nilai bobot dalam metode similaritas sangatlah mempengaruhi hasil perhitungan. Nilai bobot suatu kata diperoleh dari jumlah kata dibagi dengan total kata yang ada, sehingga semakin banyak total kata maka semakin kecil pula nilai bobot yang akan diperoleh untuk suatu kata. Berdasarkan hasil grafik pengujian, metode *AND* memiliki sifat membatasi hasil pencarian, karena hasil yang akan ditampilkan hanyalah hasil yang sesuai dengan semua *input*-an yang ada, jika satu *input*-an tidak ditemukan maka hasil tidak ditemukan. Metode *OR* sifatnya melebar, karena hasil yang ditampilkan hanya jika satu *input*-an saja ditemukan. Hal ini akan membuat halaman *web*

yang ditampilkan menjadi banyak dan isinya melebar. Metode similaritas berada di antara metode *AND* dan metode *OR*, sehingga dapat disimpulkan bahwa hasil pencarian dengan metode similaritas sifatnya tidak terlalu lebar maupun tidak terlalu sempit. Pencarian dengan metode similaritas membatasi hasil yang ditampilkan tidak terlalu banyak, dengan kata kunci yang diinputkan semakin banyak. Terlihat dari hasil analisis, pada saat 12 kata kunci diinputkan hasil yang didapatkan semakin spesifik, berbeda dengan metode *AND* yang nihil hasil dan metode *OR* yang memberikan semua dari kasus yang digunakan.

Daftar Pustaka

- [1] Ellitan, Lena. 2002. Praktik-praktik Pengelolaan Sumber Daya Manusia dan Keunggulan Kompetitif Berkelanjutan. *Jurnal Manajemen & Kewirausahaan*. 4(2) : 65-76.
- [2] Hasugian, Jonner. 2006. Penelusuran Informasi Ilmiah Secara Online : Perlakuan Terhadap Seorang Pencari Informasi Sebagai Real User. *Pustaka : Jurnal Studi Perpustakaan dan Informasi*. 2(1) : 1-13.
- [3] Asian J., Williams H. E. dan Tahaghogi, S.M.M.. 2005. Stemming Indonesian, Melbourne, RMIT University.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.59.4851&rep=rep1&type=pdf> (diakses tanggal 12 Juni 2010)
- [4] Agusta, Ledy. 2009. Perbandingan Algoritma *Stemming* Porter dengan Algoritma Nazief dan Adriani untuk *Stemming* Dokumen Teks Bahasa Indonesia.
<http://yudiagusta.files.wordpress.com/2009/11/196-201-knsi09-036-perbandingan-algoritma-stemming-porter-dengan-algoritma-nazief-adriani-untuk-stemming-dokumen-teks-bahasa-indonesia.pdf>. Diakses tanggal 11 Mei 2010.
- [5] Setyawan, S.H. dan Sarno. 2005. Fuzzy Logics Incorporated to Extend Weighted-Tree Similarity Algorithm for Agent Matching in Virtual Market. *Information and Communication Technology Seminar*. 1(1) : 49-54.
- [6] Sarno, Riyanarto. 2003. *The AgentMatcher Architecture Applied to Power Grid Transactions*.
- [7] Sommerville, Ian. 2004. *Software Engineering 7th Edition*. United States of America : Addison Wesley.